

REMARKS

No claims are amended, added, or cancelled. Thus, Claims 16-22, 38-44, 60-66, and 82-88 are pending in the application.

I. ISSUES RELATING TO PRIOR ART – SECTION 103

A. CLAIMS 16, 18-19, 21-22, 38, 40-41, 43- 44, 60, 62-63, 65-66, 82, 84-85, AND 87-88 – *GORINGE AND KEKIC*

Claims 16, 18-19, 21-22, 38, 40-41, 43- 44, 60, 62-63, 65-66, 82, 84-85, and 87-88 stand rejected under 35 U.S.C. §103(a) as unpatentable over “*Goringe*” (US Pub. No. 2003/0131096) in view of “*Kekic*” (US Pat. No. 6,664,978). The rejection is respectfully traversed.

The following is provided for understanding the claims and the cited references. Various features of various claims are described for purposes of exposition, but not for the purpose of arguing any single claim that expresses or requires that feature. The limitations of any particular claim, and distinguishing feature thereof, are explained later.

The Specification explains a context for the claimed invention. In an embodiment, certain values stored in a MIB object are used for bootstrapping communication between a management station and a managed device. If the value for a MIB object known to the management station does not match the value stored within the MIB object on the managed device, then the management station may fail to communicate with the managed device. Thus, it is important for the management station to maintain the “correct value” for certain MIB variables. A “correct value” for a MIB variable is the value stored in the MIB object. One claimed approach includes a way to resynchronize MIB object values between a management station and a managed element.

Goringe describes a discovery agent that uses various techniques to discover a set of community strings to be used for requesting services from network devices. Once discovered, the agent attempts to validate the discovered community strings. *Goringe*'s technique involves how to **discover** which community strings to validate, **and not the validation** process. Thus, information about a validation request, such as how many community strings are sent to be validated in one validation request and the information contained in the reply, is not disclosed.

Kekic describes a managed element server that is comprised of two independent components: a graphical user interface builder and a network manager. The user interface builder allows a user to build an element manager without writing code. The network manager provides a run-time environment for executing the element managers that monitor and manage computer network behavior. An element manager may use standard SNMP GET requests for its management function.

A standard SNMP GET request includes the name of the MIB object to access and a community string that serves as an authentication token for authorizing access rights to the MIB object. A value may be stored in a MIB variable, and this value is independent from the MIB variable name and the required authentication token. Thus, a standard SNMP GET request does not include (a) proposed value(s) to be matched against the actual MIB variable value.

Claim 16

Claim 16 recites in its entirety:

*“A method for verifying information on a managed device, comprising:
a computer system comprising a managed device performing:
receiving, from a requester that stores an incorrect attribute value for an SNMP
MIB object and that is unable to read and write the SNMP MIB object
directly, and unable to obtain MIB object specification information, and
that does not have a correct value for the SNMP MIB object, a SNMP*

GET request identifying an SNMP MIB object and also containing a plurality of non-null values comprising proposals for a correct value of the SNMP MIB object, wherein the SNMP GET request requests a determination as to whether any of the values matches the correct value stored in the SNMP MIB object of the managed device; and determining whether any of the values matches the correct value stored in the SNMP MIB object; and completing execution of the SNMP GET request by: transmitting a notification message indicating whether any of the values matches the correct value of the SNMP MIB object and without providing the correct value in response to the SNMP GET request.”

Neither *Goringe* nor *Kekic* individually or in combination, teaches or suggests

*“a SNMP GET request identifying an SNMP MIB object and also containing a **plurality of non-null values comprising proposals for a correct value of the SNMP MIB object**”*

The Office Action acknowledges that *Goringe* is silent with reference to the quoted feature and relies on *Kekic*, Col. 86, Lines 25-46, to allegedly teach this feature. However, the cited passage describes a standard SNMP GET request containing a MIB variable name and device name that identifies which MIB object's value is requested and a community string that serves as an authentication token for determining whether or not to grant access to the MIB object. As explained above, a MIB variable's name, required community string, and stored values are distinct. The Office Action appears to be relying in its rejection on the special case where the requested MIB object value is that of the community string itself, alleging that the community string in the SNMP GET request is a proposal for the correct value of the community string of a MIB object. However, the standard SNMP GET request described in *Kekic* only contains a single community string and thus does not provide a plurality of proposed values for a community string value, as claimed. Therefore, even in this special case, *Kekic* does not teach or suggest

containing a **plurality** of non-null values comprising **proposals for a correct value** of the SNMP MIB object, as claimed.

Furthermore, the presence of the community string in a standard SNMP GET request is for the purpose of authorizing the request, and not for the purpose of proposing a value for the requested variable. Thus, even in *Kekic's* system, if a valid SNMP GET request is issued for the value of a MIB variable containing a community string, the supplied community string value is only expected to match the value stored in the community string MIB variable when the system is configured to require the same community string value for authorization. The system might be configured to require presenting a different community string value in order to be granted access to the desired community string value. Thus, in this special case and in general, a standard SNMP GET request does not contain a proposed value for a value stored in a MIB object, as claimed.

In addition, neither *Goringe* nor *Kekic*, individually or in combination, teaches or suggests:

*“transmitting a notification message **indicating whether any of the plurality of values matches the correct value** of the SNMP MIB object”*

The Office Action acknowledges that *Goringe* does not teach this feature, and relies on *Kekic*, Col. 86, Lines 25-46 to allegedly teach this feature. As explained earlier, this passage describes a standard SNMP GET request. The Office Action appears to consider the error message generated in response to an invalid parameter to be equivalent to the claimed notification message that indicates whether any of a plurality of values matches a correct value stored in a SNMP MIB object. However, as explained earlier, a standard SNMP GET request does not contain a plurality

of proposed values stored in a SNMP MIB object, as claimed. Thus, the error message could not possibly indicate whether or not a plurality of proposed values matches the correct value, as claimed.

Applicant has identified several distinguishing features of Claim 16 not found in any combination of *Goringe* and *Kekic*. The Office has not established a *prima facie* case of unpatentability for these features or provided any evidence why a skilled person would have understood the references to provide the use of proposed MIB values. For at least these reasons, Claim 16 is patentable under 35 U.S.C. §103(a) over *Goringe* and *Kekic*. Reconsideration and withdrawal of the rejection is respectfully requested.

Claims 38, 60, and 82

Claims 38, 60, and 82 are independent claims that include all the same features as Claim 16. Claim 38 is a computer-readable medium form of Claim 16, and Claims 60 and 82 are apparatus claims containing the subject matter Claim 16. Therefore, Claims 38, 60, and 82 are patentable under 35 U.S.C. §103(a) over the combination of *Goringe* and *Kekic* for at least all the same reasons as Claim 16. Reconsideration and withdrawal of the rejection is respectfully requested.

Dependent Claims

All of the independent claims, Claims 16, 38, 60, and 82, have been shown to be patentable over the combination of *Goringe* and *Kekic*. Thus, each and every dependent claim is also patentable under 35 U.S.C. §103(a) over the combination of *Goringe* and *Kekic* by virtue of its dependency on a patentable independent claim. Reconsideration and withdrawal of the rejection is respectfully requested.

In addition, each of the dependent claims introduces one or more additional features that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those features is not included at this time.

B. CLAIMS 17, 39, 61, AND 83 – *GORINGE, KEKIC, AND CHISHOLM*

Claims 17, 39, 61, and 83 stand rejected under 35 U.S.C. § 103(a) as allegedly unpatentable over the combination of *Goringe* and *Kekic* as applied to claims 16, 38, 60, or 82 above, and further in view of “*Chisholm*” (US Pat. No. 6,697,970). The rejection is respectfully traversed.

Each of Claims 17, 39, 61, and 83 is a dependent claim that was shown above to be patentable over the combination of *Goringe* and *Kekic*. *Chisholm* does not, nor is it alleged to, disclose the features shown to be missing in *Goringe* and *Kekic* as argued above. Therefore, Claims 17, 39, 61, and 83 are each patentable under 36 U.S.C. §103(a) over the combination of *Goringe*, *Kekic*, and *Chisholm*. Reconsideration and withdrawal of the rejection is respectfully requested.

C. CLAIMS 20, 42, 64 and 86 – *GORINGE, KEKIC, AND KWAN*

Claims 20, 42, 64 and 86 stand rejected under 35 U.S.C. §103(a) as allegedly unpatentable over *Goringe* and *Kekic* as applied to claims 16, 38, 60, or 82 above, and further in view of *WhitePaper: IronShield Best Practices Hardening Foundry Routers & Switches to “Kwan.”* The rejection is respectfully traversed.

Each of Claims 20, 42, 64, and 86 is dependent on one of Claims 16, 38, 60, or 82, and thus, has been shown above to be patentable over *Goringe* and *Kekic*. *Kwan* does not, nor is it

alleged to, disclose the distinguishing features shown to be missing from *Goring* and *Kekic* as argued above. Therefore, Claims 20, 42, 64, and 86 are patentable over the combination of *Goring*, *Kekic*, and *Kwan*. Reconsideration and withdrawal of the rejection is respectfully requested.

II. CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Please charge any applicable fee that is missing or insufficient to Deposit Account No. 50-1302.

Respectfully submitted,
Hickman Palermo Truong & Becker LLP

Dated: September 22, 2009

/DeborahLCaswell#61766/
Patent Agent, Deborah L. Caswell
Reg. No. 61,766

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 754-1455
Facsimile No.: (408) 414-1076